



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escuela Técnica Superior de Ingeniería Informática
Universitat Politècnica de València

Evaluator automático de ejercicios de operaciones bitwise en lenguaje C

Proyecto Final de Carrera

Ingeniería Informática

Autor: Salvador Alemany Garrido

Directores: Antonio Martí Campoy

Ángel Perles Ivars

Septiembre de 2017

Resumen

Este proyecto final de carrera presenta la elaboración herramienta web llamada bitwise-trainer que con el propósito ayudar al aprendizaje de las operaciones bitwise con la sintaxis del lenguaje de programación C. Con ella el estudiante puede practicar ejercicios de forma autónoma e independiente de plataforma, así como estudiar su actividad. También permite a profesores ver las estadísticas de todos los usuario de la herramienta.

Palabras clave: integer, bitwise. aprendizaje, web, parsing, lógica

Tabla de contenidos

1. Introducción.....	7
1.1 Contexto.....	7
1.2 Requisitos.....	7
1.2.1 Gestión de usuarios.....	7
1.2.2 Práctica de ejercicios.....	8
1.2.3 Estadísticas.....	10
2. Diseño.....	11
2.1 Casos de uso.....	11
2.2 Dependencias en el modelo.....	13
2.2.3 Evaluador de ejercicios.....	14
3. Documentación de la herramienta.....	19
3.1 Manual de usuario.....	19
3.1.1 Página de inicio y barra lateral.....	19
3.1.2 Páginas de entrenamiento.....	20
3.1.3 Ver estadísticas personales / perfil de usuario.....	26
3.1.4 Página de estadísticas de todos usuarios.....	26
3.1.5 Gestión de usuarios.....	27
3.2 Manual de administración.....	27
3.2.1 Instalación y personalización.....	27
3.2.2 Personalización y django-admin.....	28
4. Conclusiones.....	30
4.1 Extensiones posibles.....	30

1. Introducción

1.1 Contexto

Las operaciones bitwise son aquellas que aplican sus operadores bit a bit sobre sus operandos. Estas operaciones son fundamentales en la ciencia de la computación ya que el almacenamiento de toda información por computadores es codificado en cadenas de bits y los procesadores soportan de forma nativa esas operaciones.

Esto hace que el estudio de las operaciones a nivel de bit sea parte fundamental de la formación de los ingenieros en informática. Y una forma de asentar esa formación es proporcionar una herramienta que permita al estudiante practicar esos ejercicios de forma autónoma y desde cualquier plataforma, así como al profesorado monitorizar su progreso.

1.2 Requisitos

Una vez planteado que el objetivo es desarrollar una herramienta que permita al estudiante practicar ejercicios de operaciones bitwise de forma independiente, la solución obvia es realizar esa Implementar dicha herramienta en forma de una aplicación web frente a una aplicación nativa nos da múltiples ventajas, tanto por movilidad ya el usuario puede practicar desde distintos dispositivos con los mismos con los mismos credenciales, manteniendo sus datos anteriores; no hay necesidad instalar aplicaciones (ni de distribuir actualizaciones), el soporte multiplataforma es más flexible tanto en escritorio como a nivel móvil ya que no hay que preocuparse de las tiendas de aplicaciones (AppStore, Google Play).

Esta aplicación web debe satisfacer una serie de requisitos.

1.2.1 Gestión de usuarios.

Los estudiantes deben poder registrarse de forma independiente en la herramienta. La web debe soportar la posibilidad de recuperar la contraseña de usuario de forma segura. Por ello, se usa el patrón de asociar el registro a una cuenta de correo electrónico, a la que se enviaría al usuario un enlace de recuperar contraseña.

De esto se deducen los siguientes casos de uso:

- Dar de alta usuario: el usuario proporciona un usuario, una contraseña y una dirección de correo electrónico.

- Iniciar sesión: el usuario introduce el usuario y contraseña. A partir de ese momento, su actividad practicando operaciones debe ser asociada a dicha cuenta.
- Cerrar sesión: el usuario termina su sesión. A partir de ese momento la actividad de entrenamiento en ese navegador no queda asociada a ninguna cuenta. Hasta que cualquier usuario inicie sesión.
- Recuperar contraseña: El proceso debe proseguir como sigue:
 1. El usuario introduce su dirección de correo electrónico.
 2. Si esta dirección está asociada a un usuario existente, se le enviará un correo electrónico con un enlace a un formulario de restablecer contraseña para el usuario en cuestión.
 3. En esa página el usuario podrá establecer una nueva contraseña.

Adicionalmente se debe permitir realizar ejercicios a usuarios anónimos que no se encuentren registrados.

Los usuarios registrados se clasifican en dos categorías estudiantes y administradores/profesores (desde ahora profesores):

1. Los estudiantes pueden acceder a toda la funcionalidad de práctica de ejercicios, y a sus estadísticas personales.
2. Los profesores acceden a todas las funcionalidades que los estudiantes pueden acceder; y adicionalmente ver estadísticas globales.

1.2.2 Práctica de ejercicios.

La herramienta debe soportar múltiples tipos de operaciones bit a bit en distintos tipos de ejercicios. No obstante la implementación de todos ellos debe seguir unas pautas comunes:

- El usuario introduce la solución en un campo de texto libre.
- Uso de la sintaxis y representaciones gráficas de los operadores típica del lenguaje C, al ser utilizada en múltiples lenguajes de programación muy implantados resulta de utilidad para los estudiantes familiarizarse con la misma.

- Los errores detectados durante la evaluación del ejercicio deben ser reportados al usuario, sin indicar como deben ser resueltos ya que esto forma parte del aprendizaje.
- Para evitar que un número finito de ejercicios limite el aprendizaje de los estudiantes, los ejercicios serán generados de forma aleatoria dados los operandos de cada tipo de ejercicio.
- Se debe ofrecer la posibilidad de obtener una solución de referencia de los ejercicios. Una vez ofrecida esa solución, la práctica de dicho ejercicio se debe detener

Distintos tipos de ejercicios deben estar soportados:

- **Establecer bits a cero:**
 1. Dado un ancho de palabra de 8 o 16 bits a elegir, se genera una palabra aleatoria y una mascara de bits que son aquellos que deben ser puestos a cero.
 2. El usuario dispondrá de un campo de texto libre en el que deberá introducir la sentencia o sentencias de código C que resulten en la puesta a 0 de los bits indicados en el enunciado del ejercicio.
 3. Cuando el usuario lo considere conveniente, solicitará la evaluación de su solución apretando un botón y recibirá el resultado de la misma en pantalla.
- **Establecer bits a uno:**
 1. Dado un ancho de palabra de 8 o 16 bits a elegir, se genera una palabra aleatoria y una mascara de bits que son aquellos que deben ser puestos a 1.
 2. El usuario dispondrá de un campo de texto libre en el que deberá introducir la sentencia o sentencias de código C que resulten en la puesta a 1 de los bits indicados en el enunciado del ejercicio.
 3. Cuando el usuario lo considere conveniente, solicitará la evaluación de su solución apretando un botón y recibirá el resultado de la misma en pantalla.
- **Control de flujo:** ejercicios en los que un flujo condicional depende del valor de un bit
 - Dado un ancho de palabra de 8 ó 16 bits, a elegir, se escoge un bit aleatorio de una palabra de dicha longitud.
 - El usuario introduce una expresión condicional que ejecute una lógica pre-introducida si se cumple la condición y otra si no se cumple.
 - Cuando el usuario lo considere conveniente, solicitará evaluación de su solución apretando un botón y recibirá el resultado de la misma en pantalla.

1.2.3 Estadísticas.

Para que el estudiante y el profesorado puedan controlar el progreso y la actividad de los estudiantes la aplicación guardará un registro de los intentos de resolver los ejercicios de los estudiantes, incluyendo si esos intentos han sido exitosos.

El registro contendrá la siguiente información:

- Estudiante que trato de resolver el ejercicio.
- Categoría del ejercicio.
- Validez de la respuesta.

Este registro no se mantendrá para sesiones anónimas.

Para visualizar estás estadísticas se dispondrá de dos vistas.

- **Vista de estudiante:** la aplicación dispone de un perfil del usuario actual, ese perfil debe mostrar el el número total de intentos por categoría de ejercicio, en total y con éxito. Así como el total de intentos y el total de ejercicios resueltos para todas las categorías.
- **Vista de profesorado:** muestra las mismas estadísticas de la vista de usuario, para todos los usuarios registrados incluidos otros administradores. Solo estará disponible para los usuarios de perfil profesor.

2. Diseño

2.1 Casos de uso

Los objetivos explicados en el apartado anterior pueden ser considerados casos de uso, aplicables para distintas combinaciones de actores. A continuación se explican con el apoyo de diagramas de casos de uso.

En primer lugar introducimos las distintas clases de usuarios que podemos encontrar en el sistema y las relaciones de herencia entre las mismas.

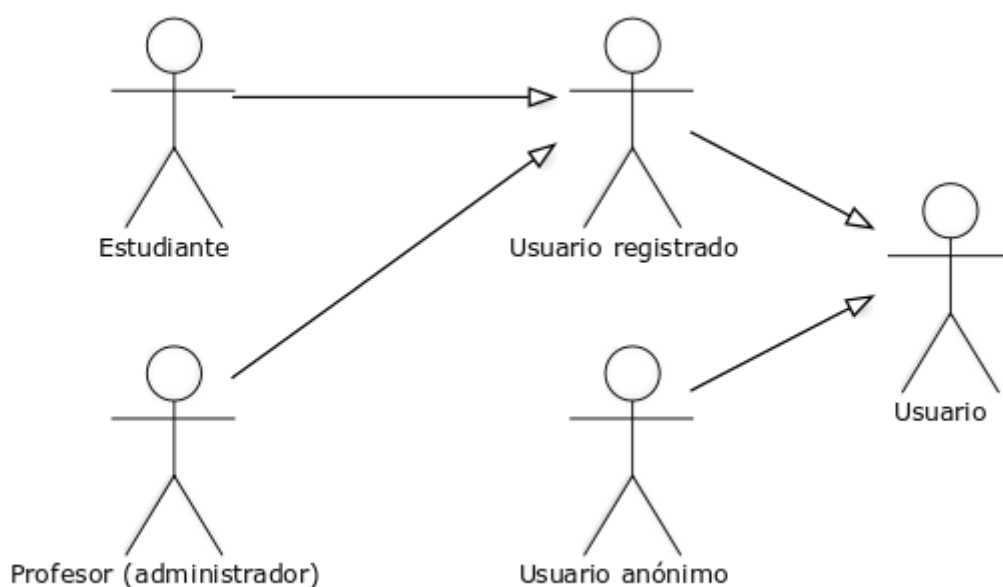


Figura 1. Diagrama de actores

- Usuario: refiere a cualquiera que interaccione con el sistema.
- Usuario anónimo: cualquier usuario que desee utilizar el sistema para practicar operaciones bit a bit sin disponer de un usuario registrado.
- Usuario registrado: cualquier usuario que disponga de una cuenta en el sistema.
- Estudiante: usuario con una cuenta en el sistema con la finalidad de aprender acerca de las operaciones bitwise

- Profesor: usuario del sistema con la responsabilidad de supervisar el aprendizaje de los estudiantes

Existen casos de uso que se considera que pertenecen a todo usuario.

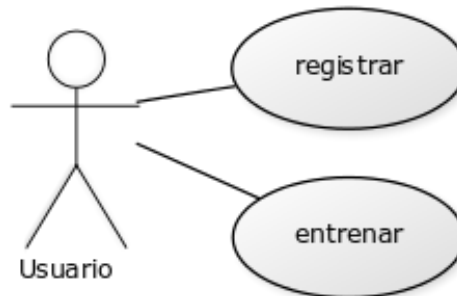


Figura 2. Casos de uso de Usuario

Todo usuario que visite la herramienta web puede entrenar sin restricciones. A si mismo el “registro “ convierte un usuario anónimo en un usuario registrado.

Aparte de entrenar un usuario registrado puede realizar las siguientes interacciones adicionales con el sistema

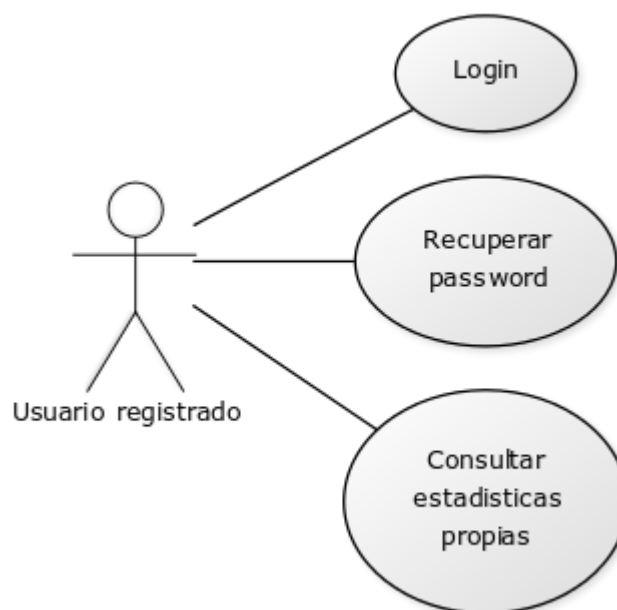


Figura 3. Casos de uso de Usuario registrado

Asimismo un administrador/profesores adicionalmente, puede consultar estadísticas de todos los usuarios:

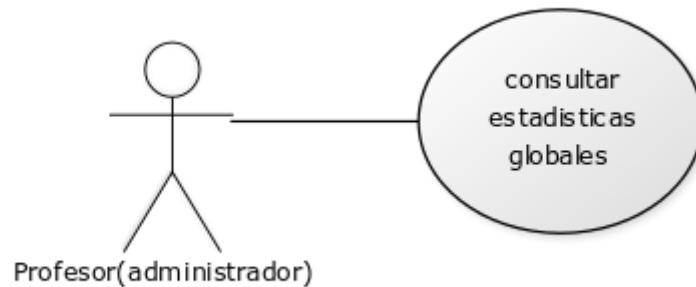


Figura 4. Ccasos de uso de Profesor(administrador)

2.2 Dependencias en el modelo

Para la implementación de la herramienta se tomaron una serie de decisiones.

Almacenar las estadísticas en una base de datos relacional.

Utilizar el framework de desarrollo web django. Proporciona una serie de herramientas que facilitan el desarrollo de la aplicación.

- Un ORM (Object-Relational Mapping): para abstraer el uso de base de datos. Los framework de mapeo objeto-relacional nos permiten convertir datos del sistema de tipos del lenguaje de programación a la base de datos usada como mecanismo de persistencia. Adicionalmente al abstraer las partes comunes entre distintas bases de datos facilita usar durante el desarrollo SQLite y soportando la migración a una base de datos más avanzada en el futuro si fuese necesario sin cambios de código.
- Un template engine: nos permite definir las distintas vistas de la herramienta con plantillas (templates).
- Una arquitectura Model-View-Template: En la que el modelo representa la capa de datos la View el acceso y modificación a esa capa de datos y la Template la vista que el usuario tiene del sistema.

Django soporta el tradicional y más implantado Python2 así como su sucesor Python3. Así pues se decidió utilizar Python3 al ser la versión recomendada para nuevos desarrollos.

La mayor parte de la lógica posible se ejecuta en el navegador de manera que resulta posible guardar una vista de entrenamiento y seguir practicando sin tener un servidor

web en marcha. Toda la generación de ejercicios, soluciones, evaluación de respuestas ha sido implementada en javascript que es ejecutado por un navegador web.

La librería de javascript jQuery ha sido por las siguientes utilidades básicas que facilitan el desarrollo de nuestra aplicación web:

- Permite una fácil manipulación y navegación del DOM (Document Object Model)
- Facilidades de manejo de eventos que permiten un uso simple de AJAX (Asynchronous Javascript And Xml), usado para almacenar los resultados de los ejercicios.

Con estas herramientas se construye una aplicación web que dispone de un conjunto de páginas que son visualizadas de forma independiente.

Adicionalmente django incluye una aplicación aplicación de administración web que con pequeñas personalizaciones ha sido adaptada para realizar tareas de administración y mantenimiento.

2.2.3 Evaluador de ejercicios.

Para el proceso de evaluación de ejercicios dado el requisito de soportar la sintaxis de C se decidió completarlo de la forma en la que tradicionalmente se implementa el soporte de un Lenguaje de Programación o un Domain Specific Language (DSL, lenguaje específico de dominio); escribiendo un pequeño compilador que acepte entradas en el subconjunto del lenguaje definido.

Salvo que sea estrictamente necesario para procesar un lenguaje con una gramática completa no se recomienda nunca escribir un *parser* del mismo manualmente ya que es tedioso, el código resultante repetitivo y difícil de modularizar y complejo de optimizar.

Adicionalmente, procesar texto dada un sintaxis y semántica definidas son problemas muy comunes; por ello existen múltiples herramientas que permiten definir la gramática en una sintaxis de alto nivel y generan un procesador del lenguaje indicado.

Dado que se requería que la evaluación de los ejercicios fuera posible estrictamente desde el navegador sin la interacción de ningún servidor web solo los generadores de parsers que soporten javascript como plataforma prioritaria eran aceptables para el proyecto.

Para ello se realizó una pequeña investigación de qué herramientas se encontraban disponibles. Se vio rápidamente que existían dos herramientas de características similares que permitirían escribir un parser para los ejercicios que se debían evaluar. Peg.js y nearley.js.

Para la selección de la herramienta definitiva finalmente se escogió peg.js. Los motivos que llevaron a esta elección son los siguientes:

- peg.js: dispone de un IDE (Entorno de Desarrollo Integrado) web completo que permite realizar todo el proceso de desarrollo y compilación de la gramática desde un navegador.
- Peg.js: utiliza una sintaxis inspirada en Flex y Bison, tradicionales estándares del procesamiento de lenguajes desde las primeras versiones en los años 1970. Por aquel entonces conocidas como Lex y yacc antes de su incorporación al proyecto GNU.
 - La diferencia más importante entre peg.js y flex+bison es la decisión de unificar las definiciones léxicas y semánticas en un mismo fichero con sintaxis unificada, simplificando la escritura y compilación de los parsers.

Las funcionalidades avanzadas, como facilidades para gestionar una tabla de símbolos e facilidades para gestionar la definición de tipos personalizados, fueron ignorados ya que se encuentran mucho más allá de la funcionalidad que se desea soportar.

Se utiliza una gramática unificada para las distintas categorías de ejercicios, definida como sigue:

```
// entry point for "set to zero" and "set to one" problems
StatementList
  = statement:Statement StatementSeparator statementList:StatementList
{
    return [statement].concat(statementList);
}
/
statement:Statement StatementSeparator _ {
    return [statement];
}

Statement
  = BinaryAssignStatement / OnesComplementStatement

// entry point for flow conditions problems
FlowExpression
  = LeftParenthesis _ inter:InternalFlowExpression _
  RightParenthesis { return inter; }

InternalFlowExpression
  = left:LeftSideExpression EqualToOperator _ num:Number {
    return {"left": left, "right": num, "operator": "Equal"};
  }
/
  left:LeftSideExpression NotEqualToOperator _ zero:"0" {
    return {"left": left, "right": 0, "operator": "NotEqual"};
```



```

}

LeftSideExpression =
    _ LeftParenthesis skip:SkippedParenExpr RightParenthesis _
{ return skip; }

SkippedParenExpr =
    id:Identifier op:BitwiseAnd num:Number
    {return {"left": id, "right": num, "operator": op}}

OnesComplementStatement
    = left:Identifier AssignOperator _ OnesComplementOperator
right:Identifier {
    return {
        "type": "OnesComplementStatement",
        "destination": left,
        "origin": right
    }
}

BinaryAssignStatement
    = id:Identifier operator:BinaryAssignOperator right:RightFactor {
        var res = {
            "type": "BinaryAssignStatement",
            "destination": id,
            "left": id,
            "right": right,
            "operator": operator.slice(0, -1)
        };
        return res;
    }

BinaryAssignOperator
    = "&=" / "|=" / "<=<=" / ">>="

BitwiseAnd
    = "&"

// for know the pure assign operator can only be used combined with
// the ones' complement operator.
// this is by design
AssignOperator
    = '='

OnesComplementOperator
    = '~'

EqualToOperator
    = '=='

NotEqualToOperator

```



```

    = '!='

RightFactor
= Number / _ OnesComplementOperator number:Number {
    return {
        "type": "OnesComplementRightFactor",
        "origin": number
    }
}

Identifier
= _ "a" _ {
    return {
        "type": "Identifier",
        "name": "a"
    };
} / _ "test" _ {
    return {
        "type": "Identifier",
        "name": "test"
    };
}

Number =
    HexInteger / DecInteger

DecInteger // underscores used to ignore whitespace
= _ [0-9]+ _ {
    return {
        "type": "Integer",
        "value": parseInt(text())
    };
}

HexInteger
= _ "0x" [0-9a-fA-F]+ _ {
    return {
        "type": "Integer",
        "value": parseInt(text())
    };
}

StatementSeparator
= ";"

_ "whitespace"
= [ \t\n\r]*

LeftParenthesis
= "("

RightParenthesis
= ")"

```

Vista esta gramática se puede observar que la salida de su ejecución no es el resultado de evaluar la lista de operaciones del usuario. La salida es una lista de diccionarios que contienen la información de la lista de operaciones bitwise que el usuario ha definido y sus operandos, en el orden en el que las mismas deben ser aplicadas al estado inicial.

Esta lista de diccionarios es leída por un modulo de la aplicación, escrito en javascript. Este modulo aplica dicha lista de operaciones a partir del estado inicial, obteniendo como resultado un “Estado final”.

Una vez hecho eso se compara el contenido del “Estado final” que debía ser modificada con la condición o condiciones que el estado debe satisfacer. En base al resultado de dicha comprobación la respuesta es considerada correcta o incorrecta. La variable de estado puede ser llamada ‘a’ o test.

Esta gramática soporta iniciar el análisis desde distintos puntos de entrada. StatementList es el punto de entrada de los ejercicios de puesta a uno y cero y FlowExpression el punto de entrada de los ejercicios de control de flujo

3. Documentación de la herramienta

3.1 Manual de usuario

3.1.1 Página de inicio y barra lateral.

Al acceder por primera vez a la herramienta web vemos la página de inicio. Para un usuario anónimo tiene el aspecto visible en la imagen a continuación.

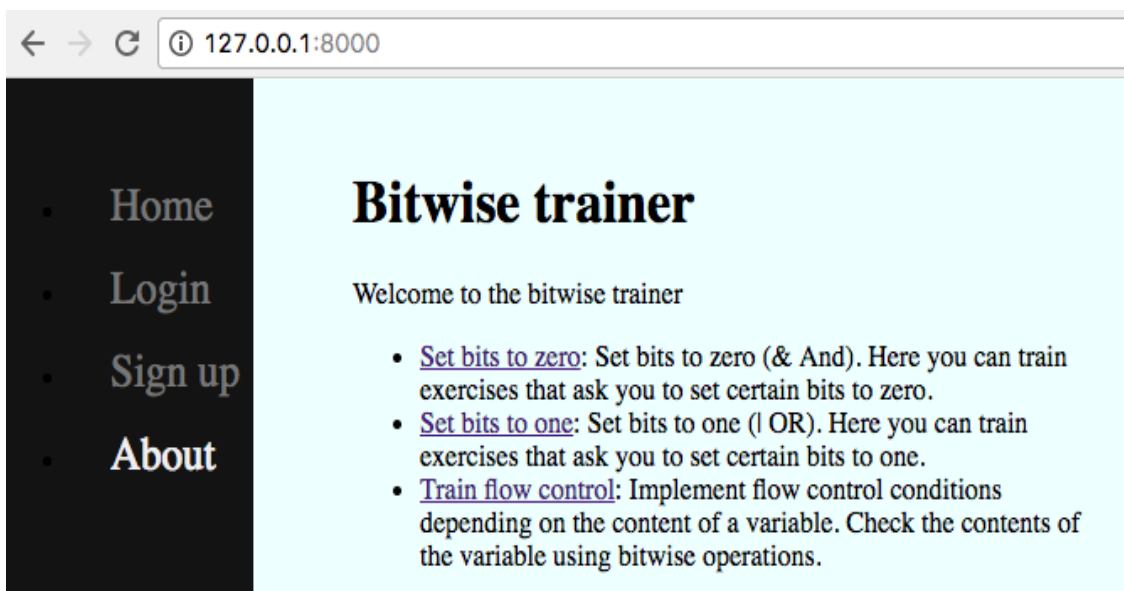


Figura 5. Página de inicio

Como podemos ver consta de dos partes, una barra lateral que contiene una lista de enlaces.

- Home: enlace a la página actual.
- Login: enlace al formulario de inicio de sesión. Para iniciar sesión cuando ya se dispone de un usuario registrado.
- Signup: enlace para registrar un nuevo usuario.
- About: Enlace a la página con información sobre la herramienta así como breves instrucciones de uso.

En la parte principal de la página home, tenemos los enlaces a las distintas categorías de ejercicios.

Cuando el usuario ha iniciado sesión, la barra lateral muestra algunas diferencias.

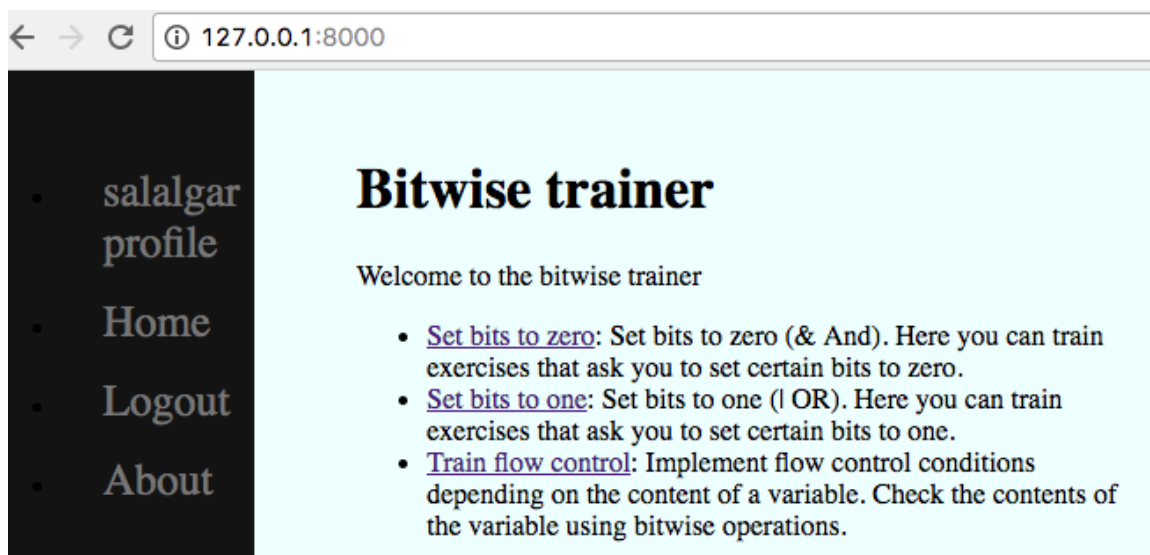


Figura 6. Página de inicio con sesión de usuario

Como se puede ver los enlaces “Login” y “Signup” desaparecen y un nuevo enlace al perfil de usuario aparece en la parte superior y un enlace “Logout” para cerrar la sesión es añadido.

Por parte de un usuario de tipo administrador la diferencia más importante es que se muestra un enlace con la etiqueta “All results” para acceder a los resultados de entrenamiento de todos los usuarios de la página web.

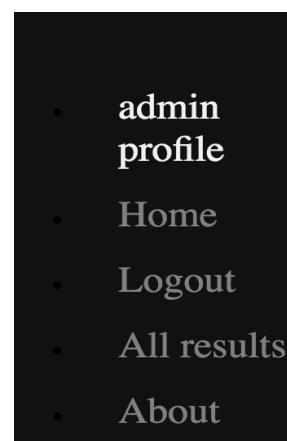


Figura 7. Barra lateral de profesor

3.1.2 Páginas de entrenamiento

La herramienta dispone de tres categorías de ejercicios:

- “Set bits to zero”: puesta a cero
- “Set bits to one”: puesta a uno
- “Flow Control”: control de flujo.

Dos categorías disponen de muchos puntos en común “Set bits to zero” (puesta a cero) y “Set bits to one” (Puesta a uno), por lo que se explican conjuntamente. El objetivo de

ambos tipos de ejercicios es asegurar que una palabra cualquiera, tiene la serie de bits establecidos por una máscara generada aleatoriamente con valores de cero o uno dependiendo del tipo de ejercicio.

Todas las categorías ofrecen la opción de intentar resolver el problema al usuario, y de mostrar una solución, después de mostrar una solución se deja de poder intentar resolver el ejercicio. Adicionalmente también se informa al usuario del resultado de su intento de resolver el problema una vez ha solicitado que se evalúe una respuesta.

La sintaxis para resolver estos enunciados es un pequeño subconjunto del lenguaje C. Con únicamente operaciones bitwise, y sin poder definir variables disponiendo únicamente de la variable de estado inicial.

Visto en imágenes.

1.0.0.1:8000/bitwise/training/set-bits-to-zero/



Category: Set bits to zero

Set bits to zero (& And).

Here you can train exercises that ask you to set certain bits to zero.

Word width

Your answer:

Figura 8. Página inicial de la categoría:

Category: Set bits to zero

Set bits to zero (& And).

Here you can train exercises that ask you to set certain bits to zero.

Word width

Bit mask that should be satisfied

<input type="checkbox"/>	<input type="checkbox"/>	0	<input type="checkbox"/>	<input type="checkbox"/>	0	<input type="checkbox"/>	<input type="checkbox"/>	0	<input type="checkbox"/>	0	0	0	0	<input type="checkbox"/>	<input type="checkbox"/>
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

Ensure using C-family bitwise operations that the variable **a** satisfies the bit mask.

Your answer:

Figura 9: Categoría con problema ya generado

A continuación se puede ver como al hacer click en “display solution” se muestra la solución:

7.0.0.1:8000/bitwise/training/set-bits-to-zero/

The screenshot shows a web application interface for a bitwise training exercise. A modal dialog box is open, displaying the solution to the problem. The modal has a title "127.0.0.1:8000 says:" and contains the text "The solution:" followed by the code "a &= ~0x24bc;". There is an "OK" button in the bottom right corner of the modal. Below the modal, the main interface shows a "Word width" dropdown set to "16" and a "Generate" button. Below that, a "Bit mask that should be satisfied" is shown as a row of 16 boxes, with bits 13, 11, 8, 5, 4, and 1 set to 0. The instruction says "Ensure using C-family bitwise operations that the variable a satisfies the bit mask." There is a text input field for "Your answer:" and two buttons at the bottom: "Evaluate" and "Display solution".

127.0.0.1:8000 says:

The solution:
a &= ~0x24bc;

OK

Here you can train exercises that ask you to set certain bits to zero.

Word width 16 Generate

Bit mask that should be satisfied

		0			0			0		0	0	0	0		
16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

Ensure using C-family bitwise operations that the variable a satisfies the bit mask.

Your answer:

Evaluate Display solution

Figura 10. Mostrando la solución

Y por último vemos como se informa de que una respuesta es correcta o incorrecta.

Your answer:

```
a &= ~0xE3;
```

Evaluate Display solution

Registered correct answer by salalgar.

Figura 11. Notificando respuesta correcta

Your answer:

```
a &= ~0xFF;
```

Evaluate Display solution

Registered wrong answer by salalgar.

Figura 12. Reportando respuesta incorrecta

Nuestro ejemplo de respuesta incorrecta establece todos los bits a cero, incluyendo los solicitados, pero establecer bits no solicitados a cero (o a uno) no es aceptable en una solución.

El problema de resolver problemas de control de flujo presenta algunas diferencias como podemos ver en la imagen a continuación:

Category: Train flow control

Implement flow control conditions depending on the content of a variable. Check the contents of the variable using bitwise operations.

Word width

Given the following code, complete the IF expression so that the variable `b` is incremented by one if the **6**-bit of the variable "test" is 1. Otherwise, `b` must be decremented by one.

Your answer:

```
if (Introduce your expression here)
{ b++;}
else
{b--;}
```

Figura 13: Enunciado problema de control de flujo

Las dos diferencias en presentación del problema son:

- En vez de mostrarse una máscara de objetivo y un estado inicial se muestra un enunciado en texto indicando el bit a comprobar en la condición.
- Se incluye un pequeño texto de inicio en la respuesta para ayudar a completar la solución.

3.1.3 Ver estadísticas personales / perfil de usuario

Si un usuario desea ver sus estadísticas o el correo electrónico con el que registró su cuenta se puede dirigir a su perfil de usuario. En el que verá esa información

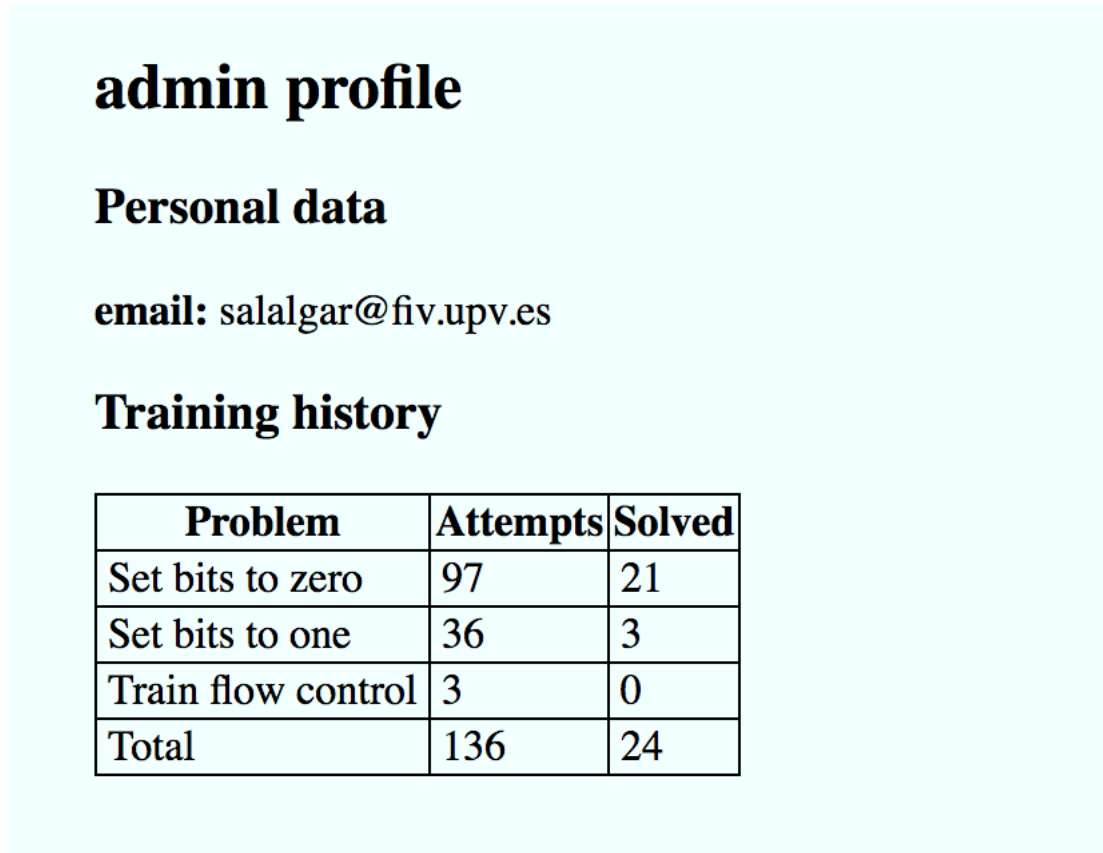


Figura 14. Perfil de usuario

3.1.4 Página de estadísticas de todos usuarios.

Esta página se muestra únicamente al administrador. Y le permite ver las estadísticas de prácticas de ejercicios para cada usuario.

Bitwise trainer

User	Set bits to zero attempts	Set bits to zero solved	Set bits to one attempts	Set bits to one solved	Train flow control attempts	Train flow control solved	Total attempts	Total solved
admin	98	21	38	3	19	9	155	33
salalgar	4	2	0	0	0	0	4	2
test1	0	0	1	1	0	0	1	1
test2	0	0	0	0	0	0	0	0

Figura 15. Estadísticas de todos los usuarios

3.1.5 Gestión de usuarios.

Para registrar un usuario se tiene que proporcionar un nombre de usuario y contraseña que cumpla las condiciones definidas en la página “Signup”.

En caso de que el usuario olvide su contraseña el enlace para re-establecer contraseña se encuentra disponible desde la página de login.

3.2 Manual de administración

3.2.1 Instalación y personalización.

La herramienta de operaciones bitwise es una aplicación web django convencional desarrollada con la versión 1.11 del framework. La instalación y personalización de las mismas se cubre extensivamente en la documentación oficial. No obstante, se prefiere proporcionar una serie de indicaciones básicas de como instalar la misma rápidamente en un ordenador local.

Para un estudiante o profesor que desea familiarizarse con la misma el proceso son unos simples pasos. En primer lugar se necesita disponer de Python3.

Python3 se encuentra en la instalación por defecto de cualquier distribución de linux de los últimos años. Si se desea instalar en Mac OS X la forma recomendada es usar la popular herramienta homebrew. El siguiente comando realizará la instalación.

```
> brew install python3
```

En caso de otras plataformas o no poder usar brew o la versión disponible en su distribución de linux dirijase a la web de Python para más información.

Una vez instalado Python necesitamos instalar las dependencias de la aplicación. Con bitwise-trainer se incluye en el directorio principal un fichero llamado requirements.txt que contiene todas las dependencias del mismo.

Para instalarlos simplemente ejecute desde una línea de comandos en el directorio donde se encuentre el proyecto:

```
> pip3 install -r requirements.txt
```

una vez la instalación se complete. Simplemente resta dirigirse al subdirectorio bitwise-trainer y ejecutar el comando

```
python3 manage.py runserver
```

La salida esperada es

Starting development server at <http://127.0.0.1:8000/>

Con estos pasos ya se dispone de una instancia de la aplicación lista para usar disponible localmente para familiarizarse con la misma.

La aplicación incluye una base de datos SQLite inicial con un usuario administrador/profesor con credenciales indicados en el fichero README.md.

3.2.2 Personalización y django-admin

Desde la interfaz de administrador de django se pueden borrar o modificar usuarios, convirtiéndolos en nuevos administradores.

Así mismo se pueden modificar los enunciados de los ejercicios. Editando los objetos de definidos en la tabla Category. Personalizaciones disponibles, editando:

- description: modificamos el enunciado mostrado.
- Initial answer: proporciona un contenido inicial para el campo de respuesta, puede ser aconsejable en algunos tipos de ejercicios.

The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000/admin/bitwise/category/8/change/`. The page title is "Django administration". The navigation bar includes "WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT". The breadcrumb trail is "Home > Bitwise > Categories > Train flow control". The main heading is "Change category". A "HISTORY" button is in the top right. The form contains the following fields:

- Name:**
- Description:**
- Slug:**
- Mode type:**
- Initial answer:**

```
if (Introduce your expression here)
{ b++;}
else
{b--;}

```

At the bottom, there are four buttons: "Delete" (red), "Save and add another", "Save and continue editing", and "SAVE".

Figura 16. Vista de edición de categoría

4. Conclusiones

Se ha desarrollado una web que proporciona al alumno la posibilidad de realizar ejercicios de entrenamiento en operaciones bit a bit.

El desarrollo de la misma me ha permitido practicar en el desarrollo de una herramienta desde cero. La herramienta reutiliza los componentes de frameworks como django o jQuery en la medida de lo posible.

El desarrollo de una gramática usando un generador de compiladores ha supuesto un reto importante que me ha permitido aprender mucho sobre como implementar herramientas que permitan procesar texto de forma compleja.

4.1 Extensiones posibles.

La herramienta se puede extender de dos maneras. Dados los ejercicios existentes se puede añadir soporte para realizarlos en palabras de distintos tamaños, actualmente solo se encuentran soportados 8 y 16 bits.

Durante el proyecto se ha escrito un procesador de un pequeño subconjunto de C. Ese procesador es completamente independiente del resto de la aplicación y puede ser utilizado y extendido para software que requiera procesar un subconjunto del lenguaje C.

Adicionalmente nuevos tipos de ejercicios pueden ser añadidos de forma relativamente fácil si el subconjunto de C requerido por ellos se encuentra soportado por el parser.

Adicionalmente una versión estática de la herramienta sin necesidad de un servidor web podría ser extraída, si se eliminan las estadísticas y la gestión de usuarios.